XLNet Summary

**What makes it different?**

* First Model to beat BERT at NLP tasks
* Achieved State-of-the-Art on 18/20 testing tasks
* General goal: combine the best of Auto-regressive (AR) and Auto-Encoding pre-training techniques (AE)

**Main Idea:**

Considers all possible permutations of the factorization order of a log sequence in its auto-regressive model. (I will give an example of this in the video). Whereas traditional AR models look only at previous tokens in the sequence and AE models randomly select out 5-15% of the tokens to be ignored while training.

Written example:

Training phrase:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 |
| Token | Let’s | get | rowdy | Roadrunners |

Classic AR, predict last two words. Model can look at previous Words:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 |
| Token | Let’s | get |  |  |

Predict rowdy at index 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 |
| Token | Let’s | get | rowdy |  |

Then predict Roadrunners at index 4.

For XLNet, randomly permute our factorization. We can only look at previous indices to make the prediction:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | 4 | 1 | 2 | 3 |
| Token |  | get | rowdy |  |

Predict Roadrunners at index 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | 4 | 1 | 2 | 3 |
| Token |  | get | rowdy | Roadrunners |

Then predict Let’s at index 4. Do this for **All permutations.**

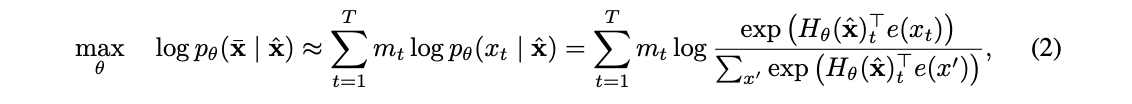
**Factorization Visualization:**

**Diagram

Description automatically generated**

**Formal definitions:**

BERT



XLNet

Diagram, schematic

Description automatically generated

**Model Architecture:**

* Paper’s model was tested with 24 layers to have a similar Architecture to BERT
* One issue the XLNet team ran into was in how to allow consideration of the previous indices method from a software engineering perspective. They devised a way that in-between layers, a given weight looks at previous weights in layers by also looking at itself simultaneously and coupling this with a gradient-independent memory component as seen in the Factorization visual.

**Other considerations:**

Cost and time of training. Being a new technique and one that involves permuting inputs, it is understandably slow and costly to train. One estimate places a quarter-million dollar price tag on its training, restricting its reproducibility.

Graphical user interface, text, application

Description automatically generated

**Results from IMDB training:**

* **Accuracy: 0.92416, eval\_loss: 0.31708**